Innovations for a Sustainable Tomorrow

LIV.INNO

FORM

Ana Costa Pereira, Josh Davies
University of Liverpool

FORM is an open-source symbolic manipulation system tailored for the physics community to perform large-scale algebraic calculations. Our work has focused on improving the parallel scalability of several benchmarks covering real-life examples of code widely used by the community. By improving the scalability of these benchmarks, we aim to make better use of modern multicore systems, reducing execution time and computational overhead, and contributing to more sustainable high-performance computing while supporting large-scale research applications.

TFORM

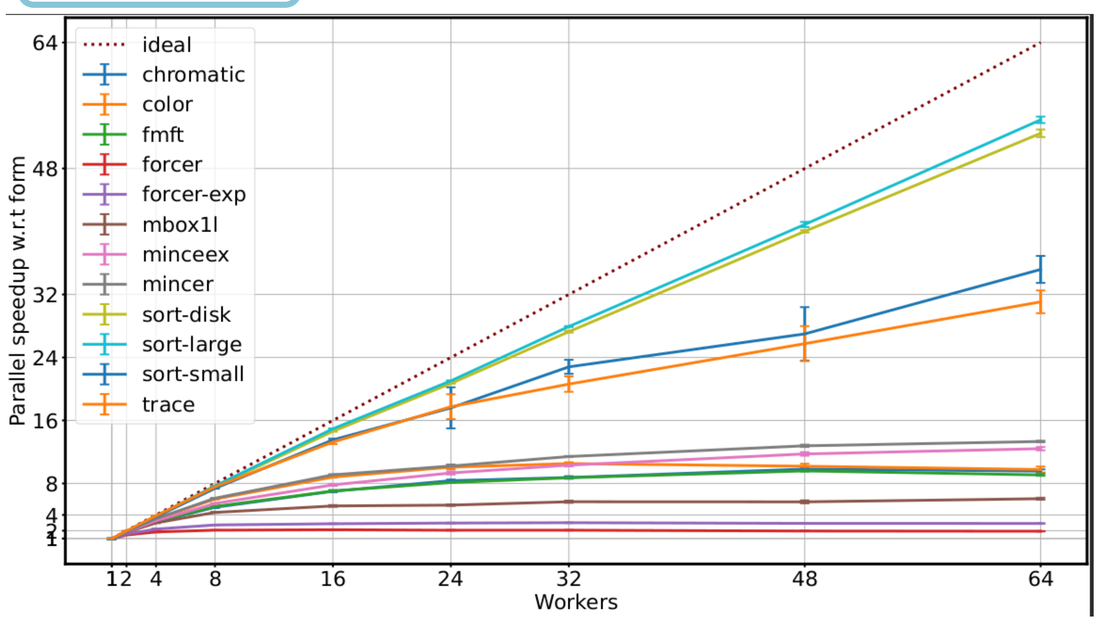


Fig1. Results for the parallel scalling of different benchmarks from COSMA8 HPC.

- sort-disk, sort-large, sort-small, trace scale up to 64;
- scaling is not good for other tests for > 16 workers.

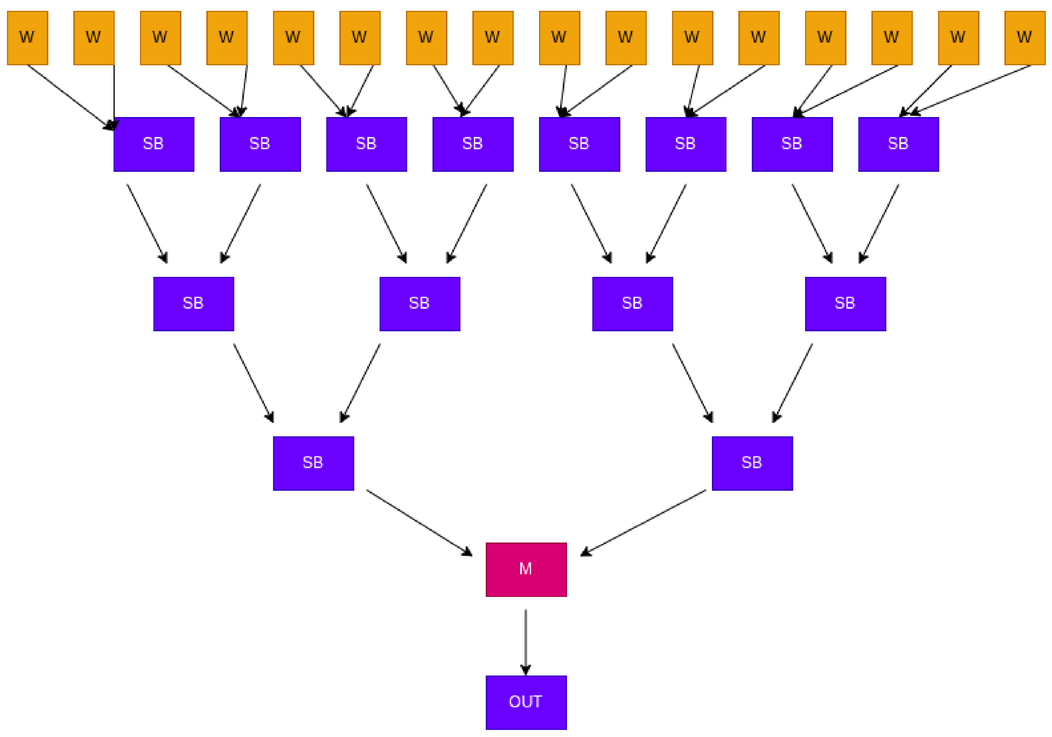
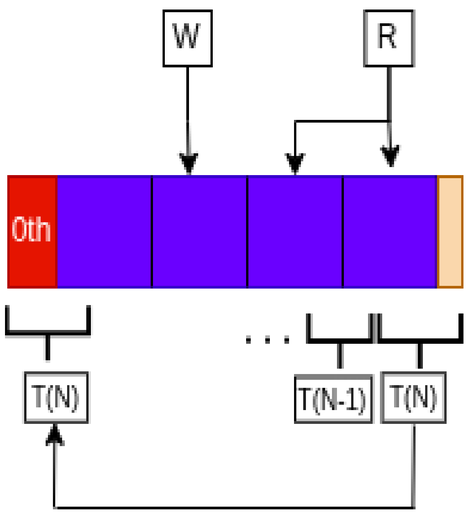


Fig2. Parallel merging of terms in the SortBot routine .

- Master → distributes terms to the workers;
- Worker→ independent stream of sorted terms;
- N workers → N − 2 SortBots are created;
- SortBot routine starts;
- Master performs the final merge and writes output.

SortBots



- SortBot → ring buffer with blocks;
- Terms are allowed to overlap between blocks;
- Writing (W) and Reading (R) threads lock active blocks.

Fig3. SortBot structure in FORM 5.0.

Bottlenecks on SortBot merging

Bottlenecks:

- Lack of parellelization at the top of the tree;
- **SortBots have to fill up to the top** before going to next with terms overlaping;
- **Large SortBlocks:** ouput from workers could fit in single block.

SortBot Changes

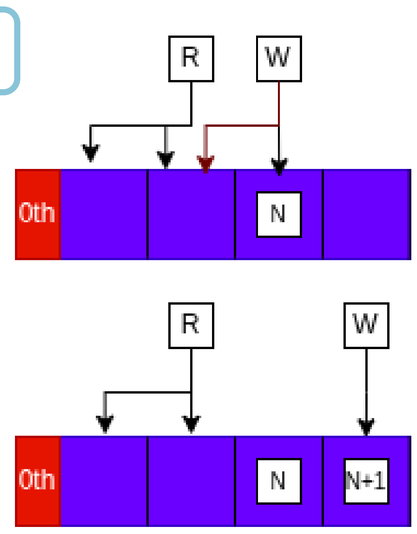


Fig4. SortBot structure after code changes.

•Require fewer SortBlocks - master buffers not expanded so often;
•Whole terms must fit into the block: if it does not fit, it goes into the next block;
•W filling the blocks tries to obtain a lock of the block before the current fill block;
•If W does not obtain the lock, R thread already has the lock – W moves to the next block.

Results

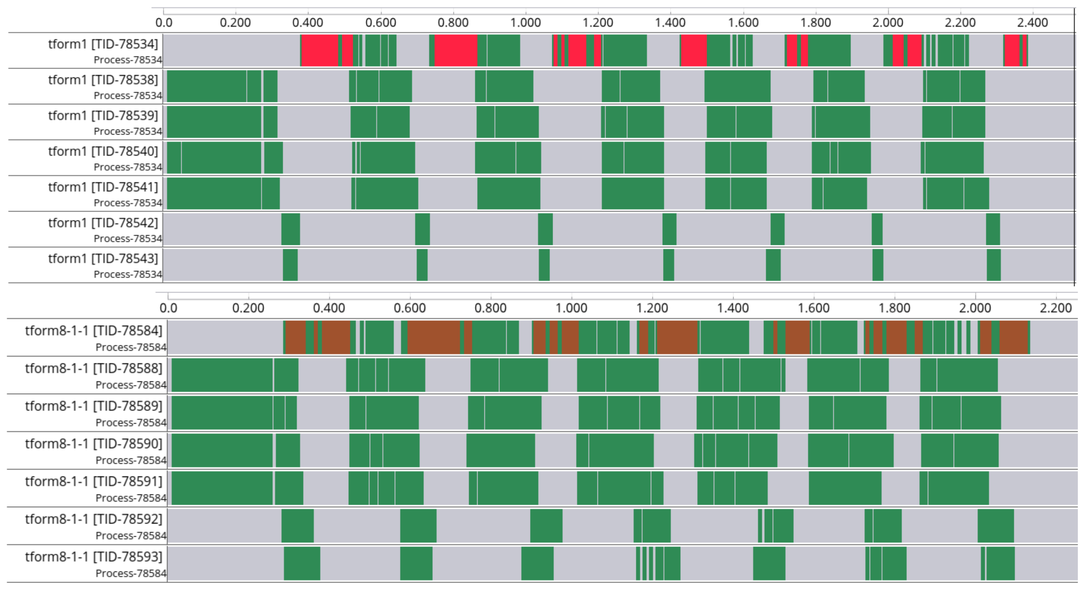


Fig5. Results from profiler AMDuProf for testcase before and after code changes.

- Master and SortBots can overlap allowing better paralelization;
- No decrease in performance in any tests;
- Forcer: 30% improvement;
- Mbox1l: 20% improvement.

This Center for Doctoral Training has received funding from the UK's Science and Technology Facilities Council.